

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE CARRERA

Ingeniería de Telecomunicación

DETECCIÓN DE RITMO CARDÍACO MEDIANTE

VÍDEO-MONITORIZACIÓN

Guillermo Guillén Mier

Junio de 2016

DETECCIÓN DE RITMO CARDÍACO MEDIANTE VÍDEO-MONITORIZACIÓN

AUTOR: Guillermo Guillén Mier

TUTOR: José M. Martínez



Video Processing and Understanding Lab

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Junio de 2016

**Proyecto parcialmente financiado por el gobierno español bajo el
proyecto TEC2014-53176-R (HAVideo)**



Resumen

El objetivo de este proyecto es adaptar y mejorar un algoritmo que nos permita detectar el ritmo cardíaco mediante el análisis de secuencias de vídeo, tanto para secuencias a color como secuencias de profundidad.

El uso de las secuencias en profundidad permite preservar la privacidad del sujeto a monitorizar y trabajar en condiciones de baja o nula iluminación.

Para obtener las secuencias, tanto en color como en profundidad, se ha diseñado un programa en C# que permite grabar dichas secuencias proporcionadas por la Kinect v2 al disco duro.

Se ha realizado un análisis del rendimiento del algoritmo en función de la distancia de la persona a la cámara con el fin de evaluar la viabilidad del mismo.

Para el análisis del algoritmo se ha grabado un conjunto de secuencias de vídeo (*dataset*) compuesto por secuencias de color y profundidad a distintas distancias junto con la grabación del ritmo cardíaco medido con un pulsómetro. Con este *dataset* se ha obtenido un conjunto de resultados en las situaciones bajo estudio.

Palabras clave

Ritmo cardíaco, video-monitorización, *dataset*, regiones de interés, análisis de secuencias de vídeo, secuencias de profundidad, seguimiento de puntos característicos.

Abstract

The objective of this Master Thesis Project is to adapt and improve an algorithm that let us detect the hearth rate through the analysis of video sequences, both color and depth.

The use of the depth sequences allow us to preserve the privacy of the subject to monitorize and to work in low or null luminosity conditions.

In order to obtain those sequences we implemented a program in C# that allow us to record the sequences provided by the Kinect v2 to the hard drive.

An analysis of the algorithm's performance has been made in order to evaluate its viability.

For the algorithm's analysis a dataset have been recorded, this dataset is composed by color and depth video sequences recorded at different distances and the hearth rate measured by a heart rate monitor. Using this dataset a complete set of results in the different situations under study has been obtained.

Key words

Heart rate, feature point tracker, region of interest, video sequence analysis, depth video sequence, detection.

Agradecimientos

En primer lugar quiero agradecer el Proyecto Fin de Carrera a mi tutor José María Martínez por el tiempo dedicado y toda la ayuda prestada. Sin olvidarme del resto de integrantes del VPU que me han ayudado cuando las cosas se torcían.

También quiero agradecer a amigos y compañeros de la carrera con los que he compartido varios años de clases, prácticas y exámenes.

Por último, pero los más importantes, a mis padres y hermanos que siempre han estado ahí desde el principio hasta el final y sin los cuales nada de esto habría sido posible.

Guillermo Guillén Mier

Junio 2016

ÍNDICE DE CONTENIDOS

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos.....	1
1.3	Organización de la memoria	2
2	Estado del arte	3
2.1	Introducción.....	3
2.2	Métodos basados en obtención del ritmo cardíaco por análisis de vídeo	3
2.2.1	Métodos basados en cambios de color	3
2.2.2	Métodos basados en cambios de movimiento	6
2.3	Conclusiones	6
3	Diseño y desarrollo	7
3.1	Introducción.....	7
3.2	Algoritmo	7
3.2.1	Algoritmo existente.....	7
3.2.2	Mejoras propuestas	12
3.3	Grabación del Dataset	14
3.3.1	Diseño.....	14
3.3.2	Desarrollo	15
4	Evaluación	19
4.1	Introducción.....	19
4.2	Marco de evaluación	19
4.2.1	<i>Ground truth</i>	19
4.2.2	<i>Dataset</i>	19
4.2.3	<i>Métrica</i>	21
4.3	Pruebas y Resultados.....	22
4.3.1	Análisis del algoritmo sobre secuencias en color del <i>dataset</i>	22
4.3.2	Análisis del algoritmo sobre secuencias de profundidad del dataset	27
4.4	Conclusiones	33
5	Conclusiones y trabajo futuro	35
5.1	Conclusiones	35
5.2	Trabajo futuro	35
6	Referencias.....	37
Anexo A: Presupuesto		39
Anexo B: Pliego de condiciones.....		41

ÍNDICE DE FIGURAS

Figura 2-1: Sistemas comerciales basados en cambios de color.....	4
Figura 2-2: Ejemplo de algoritmo basado en cambios de color con separación en bandas de color y uso de ICA. Fuente: [7].....	5
Figura 3-1: Diagrama algoritmo utilizado.....	7
Figura 3-2: Regiones de interés.....	9
Figura 3-3: Cámara Kinect v2.....	14
Figura 3-4: Resumen de las propiedades de los vídeos (color y profundidad) creados a partir de las imágenes obtenidas.....	17
Figura 4-1: Pulsómetro utilizado para obtener el Ground truth.....	19
Figura 4-2: Secuencias de la Kinect v2 que componen el dataset.....	20
Figura 4-3: Visualización de los resultados para secuencias comprimidas respecto a las secuencias sin comprimir.....	21
Figura 4-4: Visualización de los resultados para secuencias a color (DFT)...	23
Figura 4-5: Visualización de los resultados para secuencias a color (PSD)....	24
Figura 4-6: Resultados para secuencias a color (PSD vs DFT).....	24
Figura 4-7: Visualización de los resultados para secuencias a color (media fmax).....	26
Figura 4-8: Resultados para secuencias a color (fmax vs media).....	26
Figura 4-9: Comparación entre los frames de profundidad a distintas distancias.....	28

Figura 4-10: Visualización de los resultados para secuencias de profundidad (DFT).....	29
Figura 4-11: Visualización de los resultados para secuencias de profundidad (PSD).....	30
Figura 4-12: Resultados para secuencias de profundidad (PSD vs DFT).....	31
Figura 4-13: Visualización de los resultados para secuencias de profundidad (fmedia).....	32
Figura 4-14: Resultados para secuencias de profundidad (fmax vs fmedia).....	32
Figura 4-15: Visualización errores para los tres métodos utilizados.....	34

ÍNDICE DE TABLAS

Tabla 4-1: Error entre los resultados de secuencias comprimidas y sin comprimir.....	20
Tabla 4-2: Error entre los resultados de secuencias de color respecto al ground truth.....	22
Tabla 4-3: Error entre los resultados de secuencias de color respecto al ground truth utilizando la PSD.....	23
Tabla 4-4: Error entre los resultados de secuencias de color respecto al ground truth utilizando la media entre las cuatro frecuencias.....	25
Tabla 4-5: Error entre los resultados de secuencias de profundidad respecto al ground truth.....	27
Tabla 4-6: Error entre los resultados de secuencias de profundidad respecto al ground truth (DFT).....	29
Tabla 4-7: Error entre los resultados de secuencias de profundidad respecto al ground truth (PSD).....	30
Tabla 4-8: Error entre los resultados de secuencias de profundidad respecto al ground truth (media fmax).....	31
Tabla 4-9: Comparativa errores obtenidos para los distintos métodos utilizados.....	33

1 Introducción

1.1 Motivación

El ritmo cardíaco es uno de los signo vitales más importantes en el control médico de las personas. Actualmente los métodos utilizados para la medición del mismo son invasivos para el paciente pues requieren que éste vaya conectado a un sensor que realice la medición.

En la actualidad existen métodos que realizan la medición del ritmo cardíaco mediante análisis de secuencias de video a color, estos análisis se centran en la observación de las variaciones de color producidas en la piel, principalmente en la región facial. Los mayores inconvenientes de este método son no preservar la privacidad del sujeto y la necesidad de una buena iluminación en su región facial.

En este proyecto se busca adaptar y mejorar un algoritmo que permite aproximar el ritmo cardíaco del sujeto mediante micromovimientos que se producen en la cabeza debido al flujo sanguíneo. Además de trabajar con secuencias a color, se utilizarán secuencias de vídeo en profundidad.

Este algoritmo nos permite estimar el ritmo cardíaco sin utilizar dispositivos intrusivos para el sujeto además de, en el caso de la secuencia en profundidad, preservar la privacidad del mismo y permitir la grabación sin necesidad de iluminación.

1.2 Objetivos

A continuación, se enumeran y citan brevemente, las fases de trabajo para conseguir el objetivo de este proyecto:

- **Estudio del estado del arte:** Se analizarán los métodos y dispositivos de ritmo cardíaco que existen actualmente, teniendo en cuenta las ventajas y desventajas de cada uno de ellos.
- **Desarrollo de un conjunto de datos de evaluación (*dataset*):** Se grabarán videos y anotaciones del ritmo cardíaco obtenido con un pulsómetro, grabados a distintas distancias y varias modalidades. Estará constituido por secuencias a color y en profundidad grabadas con la cámara Kinect v2
- **Desarrollo de una metodología de evaluación de resultados.**
- **Adaptación del algoritmo existente previamente para su funcionamiento con secuencias grabadas con la Kinect v2.**
- **Mejora del algoritmo para secuencias en profundidad.**
- **Evaluación de la viabilidad del desarrollo de un sistema combinado en función de la distancia del sujeto a la cámara.**

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1: Introducción.** Motivación, objetivos y estructura de la memoria del proyecto.
- **Capítulo 2: Estado del arte en la obtención del ritmo cardíaco mediante análisis de vídeo.**
- **Capítulo 3: Explicación del algoritmo y obtención dataset.**
- **Capítulo 4: Descripción del dataset generado.** Descripción de las pruebas realizadas y discusión sobre los resultados obtenidos
- **Capítulo 5: Conclusiones y trabajo futuro.** Conclusiones obtenidas tras el análisis de resultados. Problemas pendientes, posibles mejoras y líneas de trabajo futuras.
- **Referencias y anexos.**

2 Estado del arte

2.1 Introducción

El estudio del ritmo cardíaco reconoce antecedentes milenarios [1] pero no se mencionan las fases de contracción y relajación ventriculares hasta 1628 en la descripción fundacional de la circulación sanguínea de William Harvey.

Después de ahondar en diversa bibliografía relacionada con la temática del proyecto podemos dividir las técnicas de obtención del ritmo cardíaco mediante análisis de vídeo en dos grandes grupos: variaciones de color y movimiento.

La obtención del ritmo cardíaco mediante variaciones de color se basan en analizar los cambios de color que se producen en la piel (principalmente en la región facial) mientras que las técnicas basadas en movimiento analizan micromovimientos que se producen en la región facial debido al flujo sanguíneo.

Ambos cambios (color y movimiento) son imperceptibles para el ojo humano, pero somos capaces de apreciarlos gracias a diversas técnicas de amplificación de vídeo [2]. Existían trabajos que se basaban en analizar los cambios de color, pero no es hasta dicha publicación que se produce un aumento considerable en la producción de trabajos basados en cambios de color y posteriormente en movimiento.

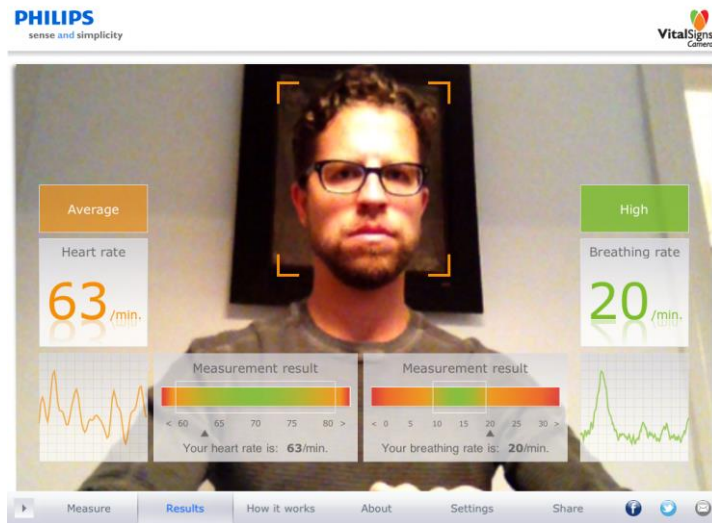
2.2 Métodos basados en obtención del ritmo cardíaco por análisis de vídeo

La obtención del ritmo cardíaco por análisis de vídeo se lleva utilizando unos años, en 2002 ya se utilizaron secuencias obtenidas por cámaras térmicas [3]. En los últimos años han surgido nuevos trabajos que utilizan cámaras convencionales, en 2007 ya se utilizaron secuencias de color con cámaras convencionales [4] pero el crecimiento empieza a producirse después de la publicación de un trabajo del Instituto Tecnológico de Massachusetts (MIT) en el que se mostraba la variación de color en el rostro producida por el flujo sanguíneo [2].

2.2.1 Métodos basados en cambios de color

La obtención del ritmo cardíaco mediante cambios de color es el método más utilizado actualmente, la técnica se basa en buscar la frecuencia a la que tenemos la máxima energía en un espacio transformado. Se obtienen las variaciones de color de los píxeles en la región facial debidos al flujo sanguíneo para filtrarlas en la banda apropiada.

Actualmente existen sistemas comerciales que utilizan este método como *Philips Vital Signs Camera* o la aplicación móvil desarrollada por *Cardiio* (ver Figura 2-1).



(a) Philips Vital Signs Camera. Fuente: Google images



(b) Cardiio app. Fuente: Google images

Figura 2-1: Sistemas comerciales basados en cambios de color

Estas aplicaciones se basan en la selección de la región de interés (rostro) mediante un algoritmo, normalmente (Viola-Jones [5]) aunque en algunos casos utilizan la transformada de Haar entrenada.

Para medir los cambios en el color se utilizan distintas técnicas:

- Comparar frames en HSI [6].
- Separar la imagen en R, G y B y normalizarlos para aplicar Análisis de Componentes Independientes (ICA, Independent Components Analysis) o Análisis de Componentes Principales (PCA, Principal Components Analysis) [7].
- Separado en R, G y B y procesado posterior sin utilizar ICA ni PCA [8].
- Realizar la operación AND de la imagen binarizada mediante la clasificación del tono de piel con el uso de la escal cromática de Fitzpatrick [9] y la imagen en el espacio YUV [10].
- Mediante el cálculo de las diferencias de brillo [11].
- Uso de descomposición piramidal y análisis de la componente R [12].

También se utilizan distintas variantes para obtener la frecuencia dominante siendo las más usadas el cálculo de la Transformada Rápida de Fourier (FFT – Fast Fourier Transform) [13] y la Densidad Espectral de Potencia (PSD – Power Spectral Density) [14].

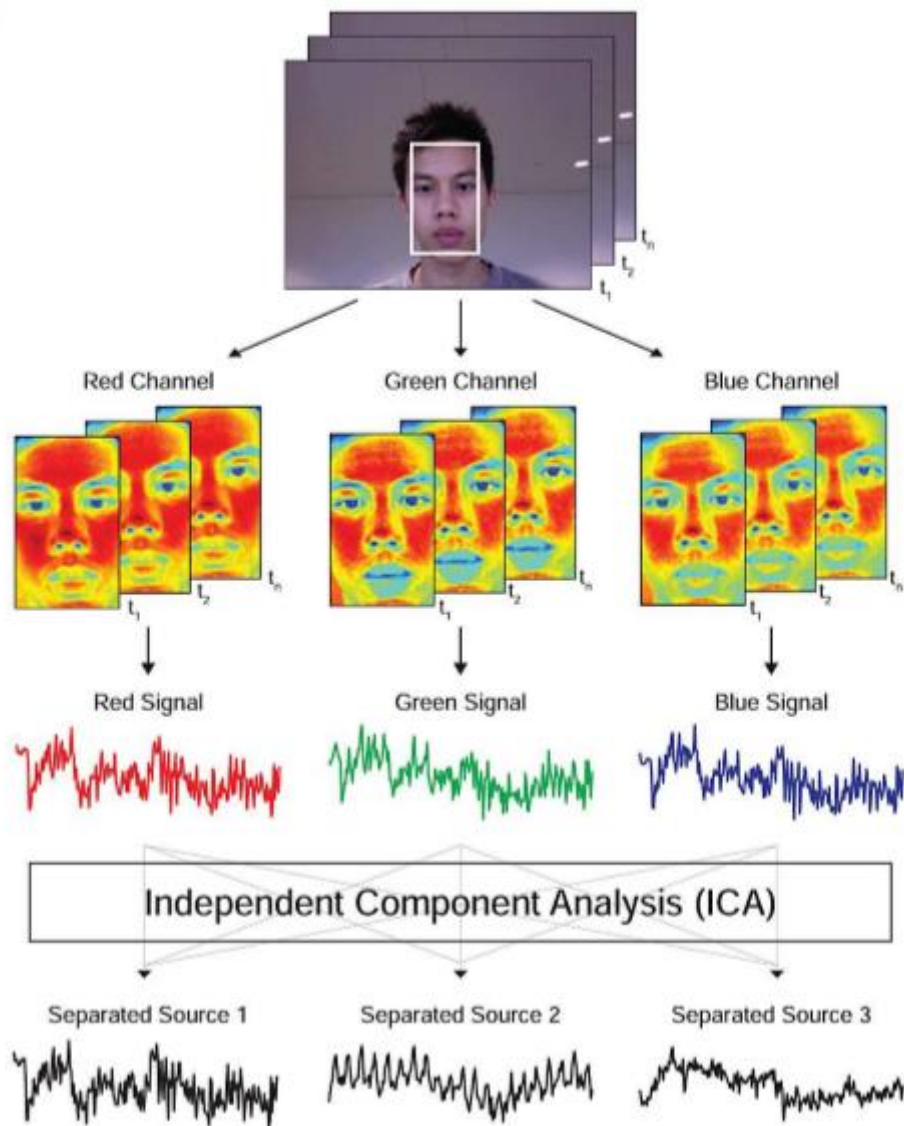


Figura 2-2: Ejemplo de algoritmo basado en cambios de color con separación en bandas de color y uso de ICA. Fuente: [7]

Estos sistemas necesitan que la piel de la zona de interés sea visible para la cámara en cambio los métodos basados en cambios de movimiento no se ven limitados por este aspecto.

2.2.2 Métodos basados en cambios de movimiento

Actualmente solo hemos encontrado tres referencias que utilicen los cambios de movimiento en el rostro para obtener el ritmo cardíaco. En 2013 Balakrishnan et al. [15] en el Instituto Tecnológico de Massachusetts (MIT) presentó el primer trabajo que utilizaba este método. Utilizaba el algoritmo de *Viola-Jones* para establecer la región de interés, para posteriormente obtener los puntos característicos de esa región con el algoritmo *Good Features to Track* [16] y realizar su seguimiento mediante el algoritmo *Kanade Lucas Tracking* (KLT) [17]. Una vez obtenida la trayectoria de los puntos se le aplica un filtrado en la banda de frecuencias adecuada, y una vez se ha filtrado la señal se realiza el análisis de las componentes principales y sobre su resultado se aplica la Transformada Discreta de Fourier para obtener el ritmo cardíaco.

En 2014, en la Universidad de Aalborg, Irani et al. [19] replicaban el algoritmo realizado en el MIT y proponían modificaciones para el mejor rendimiento del algoritmo cuando el sujeto movía voluntariamente la cabeza. Los cambios introducidos fueron el uso de un filtrado de suavizado después del filtrado de la señal y el uso de la Transformada Discreta del Coseno en lugar de la Transformada Discreta de Fourier.

En 2015 Erik Velasco Salido en su Trabajo de Fin de Grado [19], utilizó el algoritmo propuesto por el MIT y le aplicó mejoras para poder utilizar las secuencias de imagen en profundidad proporcionadas por la Kinect.

2.3 Conclusiones

Se ha decidido trabajar sobre la tecnología basada en los cambios de movimiento para hallar el ritmo cardíaco por ser una tecnología novedosa y menos estudiada hasta el momento, además podemos estimar el ritmo cardíaco con imágenes en profundidad que preservarán la privacidad del sujeto así como la posibilidad de analizar al sujeto en condiciones de baja o nula luminosidad.

Como valor añadido, al disponer de la Kinect v2 podremos obtener imágenes en profundidad con muy buena resolución lo que es posible que nos proporcione mejoras considerables en los resultados de los algoritmos a utilizar: el algoritmo del MIT implementado como base en [19].

3 Diseño y desarrollo

3.1 Introducción

Analizado el estado del arte, se utilizará una reproducción del algoritmo desarrollado por Balkrishnan en el MIT [15], si bien puede haber partes que no estén implementadas de igual forma por falta de información en el artículo.

Este algoritmo se implementó en Matlab en 2015 por Erik Velasco Salido [19]. Adicionalmente se implementaron mejoras sobre el algoritmo por su mal funcionamiento sobre las secuencias en profundidad obtenidas con la Kinect, pero nosotros utilizaremos el algoritmo base pues, con el cambio de *hardware* (Kinect v2), dichas modificaciones no son necesarias debido a la calidad aumentada en las secuencias de profundidad.

Se ha diseñado y desarrollado un programa en C# para poder grabar las distintas secuencias de video que proporciona la Kinect v2.

3.2 Algoritmo

3.2.1 Algoritmo existente

El algoritmo se basa en la detección del ritmo cardíaco analizando los micromovimientos que se producen en la región facial debido al flujo sanguíneo, para ello, se ha seguido el esquema de la Figura 3-1.

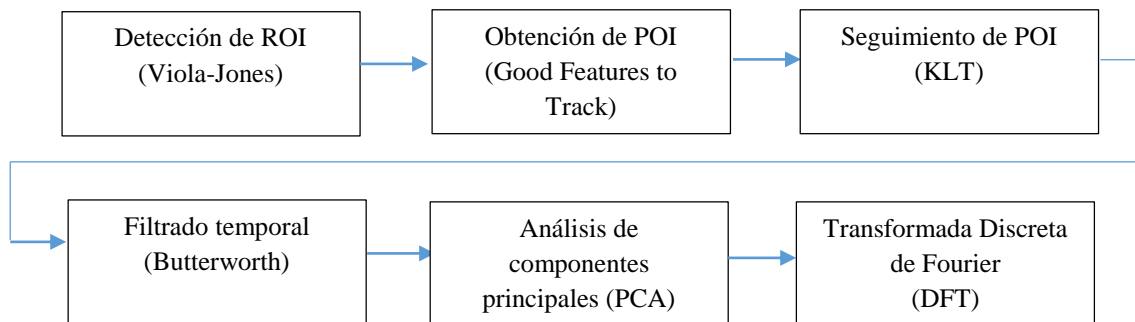


Figura 3-1: Diagrama algoritmo utilizado

Primero se obtiene la región de interés (región facial) con el algoritmo de Viola-Jones [6], una vez seleccionada la región de interés se obtienen los puntos de interés dentro de esa región mediante el algoritmo Good Features to Track [16].

Sobre los puntos de interés se realiza el seguimiento de su trayectoria con el algoritmo KLT [17]. Las señales obtenidas del seguimiento se filtran con un filtro paso banda Butterworth de orden 5 y banda de paso entre 0.75 Hz y 5 Hz. Una vez filtradas se les aplica el análisis de componentes principales (PCA) quedando sólo las componentes en el rango comprendido entre la segunda y la quinta. Para terminar, se realiza la

Transformada Discreta de Fourier (DFT) y se calcula su energía para buscar la frecuencia en la que es máxima y con este resultado se calcula el ritmo cardíaco.

El algoritmo está implementado íntegramente en Matlab. La función principal del algoritmo tiene dos opciones de ejecución, utilizar el vídeo completo y calcular el ritmo cardíaco de la secuencia completa o dividirlo en ventanas de tamaño a elegir para calcular el ritmo cardíaco por ventana. La idea detrás de enventanar la secuencia es principalmente el poder ir mostrando en “tiempo real” el ritmo cardíaco que se está monitorizando.

A continuación, se muestra el pseudocódigo de la función principal del algoritmo:

```

1  %% Main algoritmo base
2  %% Entrada: nombreFichero - vídeo para procesar;
3  % nFrames - número de frames del video
4  % use_window - parámetro que determina si usamos enventanado;
5  % t_window - tamaño de la ventana en segundos
6  %% Salida: HR - ritmo cardíaco medio medido por el algoritmo
7  function [HR]=algoritmo_base(nombreFichero, nFrames, use_window,t_window)
8  -   orden=8; fl=0.75;   fh=5;   fs=30; % Establecemos los parámetros del filtro
9  -   NFFT=8192;
10 -   video=lectura_fichero(nombreFichero,nFrames); % Función que lee la secuencia a procesar.
11
12 -   [ROIAdap]=findROI(video); % Función que busca las ROI.
13 -   [POI]=findPOI(ROIAdap,video); % Función encargada de la búsqueda de los POI.
14 -   [POI_Track]=KLT(POI, video); % Función que realiza el seguimiento de los POI.
15 -   [pointT_filter] = filter_butterworth(POI_Track, orden, fl, fh, fs); % Función que filtra las t
16 -   [PCA_results]=calc_pca(pointT_filter); % Función que realiza el PCA.
17 -   [DFT_results]=calc_DFT(PCA_results, NFFT); % Función que realiza el cálculo de la DFT.
18 -   [HR]=find_HR(DFT_results,NFFT); % Función que nos devuelve el ritmo cardíaco.
19 -   end

```

En esta función se ha suprimido la parte encargada de enventanar el video para reducir su tamaño y poder explicar los pasos principales del algoritmo. Como podemos ver, lo primero es la llamada a la función *findROI* que es la encargada de seleccionar las regiones de interés del video pasado como argumento.

Se ha utilizado la función *vision CascadeObjectDetector* proporcionada por Matlab para la búsqueda de la región facial. Una vez detectada la cara se ajusta para obtener las dos ROI como se explica en el trabajo de Balakrishnan [15].

Las regiones utilizadas son las correspondientes con la frente y la región entre la nariz y la boca (ver Figura 3-2), se han elegido por ser las más adecuadas para un sujeto colaborativo que está posando y eliminan la sección ocular para que el cálculo del ritmo cardíaco no se vea afectado por el parpadeo del sujeto.

Cuando se utilizó la Kinect para realizar el cálculo [19] había secuencias de vídeo en las que las regiones de interés se habían de seleccionar manualmente, esto ya no es necesario debido al aumento de la calidad de las secuencias de video por el uso de la Kinect v2.



Figura 3-2: Regiones de interés

```
1  %% Función para la obtención de las ROI
2  %% Entrada: video - video
3  %% Salida: ROIAdap - matriz con las dos ROI
4  function [ROIAdap]=findROI(video)
5      objectFrame = video(1).frame;
6      faceDetector=vision.CascadeObjectDetector(); %Implementación de Matlab para hacer uso de Viola-Jones.
7      faceDetector.MergeThreshold=200;
8      faceDetector.ScaleFactor=1.01;
9      %Deteccion de ojos
10     SI=size(objectFrame);
11     Region = step(faceDetector, objectFrame); % Obtenemos la ROI de la cara con Viola-Jones.
12     i=195;
13     while isempty(Region)==1
14         faceDetector.MergeThreshold=i;
15         Region = step(faceDetector, objectFrame);
16         i=i-5;
17     end
18     if size(Region,1)>1
19         Region(:, :)=Region(1,4);
20     end
21
22
23     ROIAdap=AdaptROIBase(Region); % Función que nos devuelve las dos secciones de interés en una matriz.
24 end
```

Se utiliza la función *findPOI* para la detección de los puntos de interés dentro de las dos regiones ya seleccionadas. Para ello se ha utilizado la función proporcionada por Matlab *vision.CornerDetector* basada en Good Features to Track [16] siguiendo lo ya explicado y expuesto en [15]. En esta función se buscan los puntos más significativos ubicados dentro de las regiones de interés, la función devuelve las coordenadas x e y de cada uno de los puntos de interés.

```

1  %% Función para la búsqueda de los POI
2  %% Entrada: video - secuencia de video; ROIAdap - Matriz con las dos regiones de interés
3  %% Salida: POI - puntos de interés obtenidos
4  function [POI]=findPOI(ROIAdap,video)
5      objectFrame = video(1).frame;
6      CornerDetector = vision.CornerDetector('Method','Minimum eigenvalue (Shi & Tomasi)');
7      if ROIAdap(1,4)<7
8          CornerDetector.NeighborhoodSize=[3 3];
9      else
10         CornerDetector.NeighborhoodSize=[5 5];
11     end
12     CornerDetector.CornerThreshold=0.00001;
13     CornerDetector.SmoothingFilterCoefficients=fspecial('log', [1 9], 0.75);
14     CornerDetector.MaximumCornerCount=750;
15
16     foreheadPoints = step(CornerDetector, rgb2gray(imcrop(objectFrame, ROIAdap(1,:)))); % Llama
17     foreheadPoints = double(foreheadPoints);
18
19     CornerDetector = vision.CornerDetector('Method','Minimum eigenvalue (Shi & Tomasi)');
20     if ROIAdap(2,3)<7
21         CornerDetector.NeighborhoodSize=[3 3];
22     else
23         CornerDetector.NeighborhoodSize=[5 5];
24     end
25     CornerDetector.CornerThreshold=0.00001;
26     CornerDetector.SmoothingFilterCoefficients=fspecial('log', [1 9], 0.75);
27     CornerDetector.MaximumCornerCount=750;
28
29     subnosePoints = step(CornerDetector, rgb2gray(imcrop(objectFrame, ROIAdap(2,:)))); % Llamad
30     subnosePoints = double(subnosePoints);
31
32     POI=AdaptPoints(foreheadPoints, subnosePoints, ROIAdap); % Llamada a la función que nos ad
33 end

```

Una vez detectados los puntos de interés se ha de realizar un seguimiento de los mismos obteniendo su variación a lo largo del tiempo en ambas dimensiones ($x(t)$, $y(t)$). Solamente se ha utilizado la variación de la componente en la dimensión y , desechando la componente x como se menciona en [15].

Para realizar el seguimiento se ha utilizado la función proporcionada por Matlab *vision.PointTracker*.

```

1  %% Función para el tracking de los POI
2  %% Entrada: video - secuencia video; POI - puntos de interés a seguir
3  %% Salida: POI_Track - Trayectoria de los puntos seguidos.
4  function [POI_Track]=KLT(POI, video)
5      objectFrame = video(1).frame;
6      pointTracker= vision.PointTracker('MaxBidirectionalError', 2); % Configuración del algoritmo KLT
7      initialize(pointTracker, POI, objectFrame); % Inicialización del tracker
8      for i=2:size(video,2) % Bucle para seguimiento de los puntos
9          [Points] = step(pointTracker, video(i).frame); % Llamada al algoritmo KLT
10         PointsT(i,:)=Points;
11     end
12     POI_Track=component_Y(PointsT); % Función que nos deja solamente la componente y de los puntos
13 end

```

Una vez tenemos los vectores con la variación temporal de los puntos de interés se aplica un filtro Butterworth de orden 5 con frecuencias de corte 0.75 y 5 Hz correspondientes al intervalo de 45 a 300 pulsaciones por minuto como se ha visto en [15]. Con este filtrado se eliminan variaciones causadas por movimientos externos.

```

1  %% Función que filtra la trayectoria de los puntos de interés
2  %% Entrada: POI_Track - trayectoria de los puntos seguidos; fs - frecuencia de muestreo
3  %% Salida: pointT_filter - trayectoria filtrada
4  function [pointT_filter] = filter_butterworth(POI_Track, orden, fl, fh, fs)
5      d = fdesign.bandpass('N,F3dB1,F3dB2', orden, fl, fh, fs);
6      Hd = design(d,'butter'); % Generamos un filtro Butterworth paso banda de orden 5 con banda de pasa de 0.75 a 5 HZ
7      pointT_filter=filter(Hd,POI_Track); % Filtramos las trayectorias con el filtro Butterworth
8
9  end

```

El resultado de las señales filtradas se utiliza para realizar el Análisis de Componentes Principales (PCA), se ha utilizado la función proporcionada por Matlab *pca*. Las componentes principales obtenidas se utilizan para descomponer el movimiento de los puntos característicos en subseñales para aislar el pulso. Como se ha visto en [15] se han elegido las componentes de la segunda a la quinta.

```
1 %% Función que realiza el PCA
2 %% Entrada: pointT_filter - trayectoria filtrada de los puntos
3 %% Salida: PCA_results - ritmo cardíaco obtenido con el algoritmo
4 function [PCA_results]=calc_pca(pointT_filter)
5 -     ev=pca(pointT_filter,'NumComponents',5); % Aplicamos PCA sobre la matriz de traye
6 -     signals=pointT_filter*ev; % Calculamos las señales resultantes de aplicar el PCA
7 -     PCA_results=signals(:,2:5); % Nos quedamos con las componentes 2ª, 3ª, 4ª y 5ª.
8 - end
```

Se aplica la Transformada Discreta de Fourier (DFT) con la función proporcionada por Matlab *fft* a las cuatro componentes obtenidas anteriormente.

```
1 %% Función el cálculo de la DFT
2 %% Entrada: PCA_results - trayectoria filtrada de los puntos; NFFT - número de pun
3 %% Salida: DFT_results - Matriz con las DFT de las cuatro componentes
4 function [DFT_results]=calc_DFT(PCA_results, NFFT)
5 -     for i=1:4
6 -         DFT_results(i,:)=fft(PCA_results(:,i),NFFT)/(NFFT-1); % Calculamos la DFT
7 -     end
8 - end
```

Como último paso para calcular el ritmo cardíaco, obtenemos la energía de cada una de las DFT y buscamos el máximo absoluto entre ellas para obtener la frecuencia dominante que utilizaremos para calcular el ritmo cardíaco.

Para la obtención del pulso en pulsaciones por minuto, se multiplica la frecuencia adquirida por 60 segundos/minuto obteniendo así el ritmo cardíaco [15].

```
1 %% Función que nos devuelve el ritmo cardíaco
2 %% Entrada: DFT_results - DFT de las 4 componentes resulta
3 %% Salida: HR - ritmo cardíaco obtenido con el algoritmo
4 function [HR]=find_HR(DFT_results,NFFT)
5 -     f_max=find_max_energy(DFT_results, NFFT); % Llamamos a
6 -     HR=f_max*60; % Obtenemos el ritmo cardíaco multiplicar
7 - end
```

3.2.2 Mejoras propuestas

Con el fin de intentar obtener resultados más exactos, se ha cambiado la función que calcula las Transformadas Discretas de Fourier (DFT) por una función equivalente que calcula la Densidad Espectral de Potencia (PSD) ya que hay varios trabajos que la utilizaban y se han constatado situaciones en las que la frecuencia correcta era enmascarada en un armónico al utilizar la DFT [19].

```
1  %% Función el cálculo de la PSD
2  %% Entrada: PCA_results - Componentes principales; NFFT - número de p
3  %% Salida: PSD_results - Matriz con las Densidades Espectrales de Pot
4  function [PSD_results]=calc_PSD(PCA_results, NFFT);
5  for i=1:4
6      PSD_results(i,:)=periodogram(PCA_results(:,i),[],NFFT); % Cal
7  end
8  end
```

Debido a este cambio también debemos adaptar la función que calcula el ritmo cardíaco para adecuarlo al uso de la PSD.

```
1  %% Función nos devuelve el ritmo cardíaco
2  %% Entrada: PSD_results - PSD de las 4 componentes re
3  %% Salida: HR - ritmo cardíaco obtenido con el algori
4  function [HR]=find_HR_new(PSD_results,NFFT)
5      f_max=find_max(PSD_results, NFFT); % Llamamos a l
6      HR=f_max*60; % Obtenemos el ritmo cardíaco multip
7  end
```

```
1  %% Función que nos devuelve la frecuencia a la cual tenemos el máximo en la PSD
2  %% Entrada: PSD_results - PSD calculadas; NFFT - número de puntos de la PSD.
3  %% Salida: f_max - frecuencia a la que tenemos el máximo.
4  function [f_max]=find_max(PSD_results, NFFT)
5      %Calculamos el eje de frecuencias
6      f=30/2*linspace(0,1,NFFT/2+1);
7      %Buscamos la maxima energia en cada una de las señales
8      %transformadas
9      m(1)=max(PSD_results(1,1:NFFT/2+1));
10     m(2)=max(PSD_results(2,1:NFFT/2+1));
11     m(3)=max(PSD_results(3,1:NFFT/2+1));
12     m(4)=max(PSD_results(4,1:NFFT/2+1));
13     index_m=find(m==max(m));
14     %Buscamos la posición del máximo
15     [x,index_f]=find(PSD_results(index_m,1:NFFT/2+1)==max(PSD_results(index_m,1:NFFT/2+1)));
16     f_max=f(index_f);
17 end
```

Otra mejora propuesta ha sido utilizar una media de las frecuencias obtenidas con los cuatro máximos de la señal transformada en lugar de utilizar únicamente la frecuencia a la que la energía de la transformada de la señal es máxima.

Para ello se ha modificado la función *find_max_energy* al igual que se hizo con la utilización de la PSD pero esta vez devolvemos la frecuencia media entre las cuatro frecuencias obtenidas correspondientes a la máxima energía de cada señal.

```
15 -         index_m=1;
16 -         [x,index_f]=find(real(DFT_results(index_m,1:NFFT/2+1))
17 -         f_max1=f(index_f);
18 -
19 -         index_m=2;
20 -         [x,index_f]=find(real(DFT_results(index_m,1:NFFT/2+1))
21 -         f_max2=f(index_f);
22 -
23 -         index_m=3;
24 -         [x,index_f]=find(real(DFT_results(index_m,1:NFFT/2+1))
25 -         f_max3=f(index_f);
26 -
27 -         index_m=4;
28 -         [x,index_f]=find(real(DFT_results(index_m,1:NFFT/2+1))
29 -         f_max4=f(index_f);
30 -
31 -         f_max=( (f_max1+f_max2+f_max3+f_max4)/4);
32 -     end
```

3.3 Grabación del Dataset

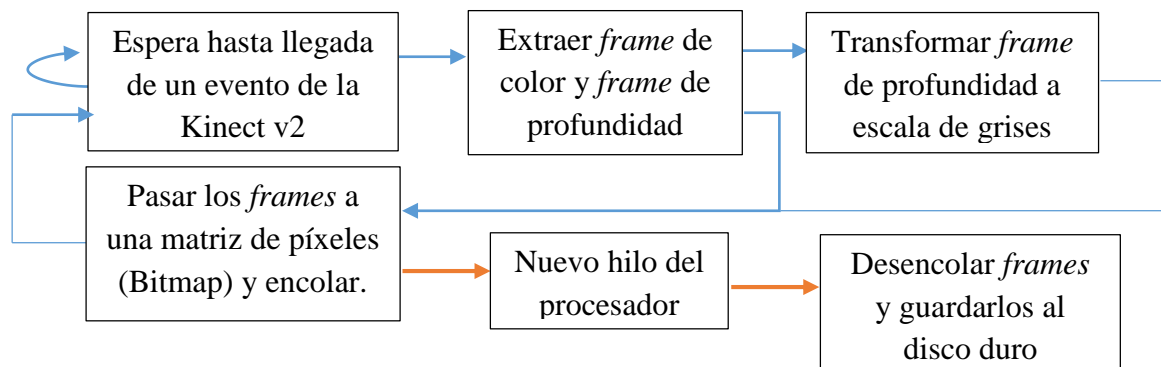
3.3.1 Diseño

Para la grabación del *dataset* se ha utilizado la Kinect v2 utilizando un adaptador USB para conectarla al ordenador. La cámara se ha colocado en la mesa de forma que el sujeto pueda posar para la grabación de las secuencias a distintas distancias (de 0.5m a 5m con intervalos de medio metro) ver Figura 3-3.



Figura 3-3: Cámara Kinect v2

Debido a la ausencia de aplicaciones de escritorio y de soporte para Matlab, se ha implementado un programa en C# para obtener las secuencias de imágenes tanto a color como en profundidad y grabarlas en el disco duro, para ello se ha seguido el siguiente esquema:



Las imágenes proporcionadas por la Kinect v2 tienen una resolución de 1920x1080 para las de color y de 512x424 para las de profundidad. También proporciona secuencias de infrarrojos e información del esqueleto que no se han grabado.

3.3.2 Desarrollo

El programa, una vez ejecutado, activa el sensor de la Kinect v2 y establecemos el método que activará un nuevo evento:

```
_sensor = KinectSensor.GetDefault(); // Accedemos al sensor predeterminado

_sensor.ColorFrameSource.OpenReader();
_sensor.DepthFrameSource.OpenReader();

timer.Start();
if (_sensor != null)
{
    _sensor.Open(); // Necesitamos abrirlo para tener acceso al sensor
    _sensor.ColorFrameSource.CreateFrameDescription(ColorImageFormat.Rgba);

    _reader = _sensor.OpenMultiSourceFrameReader(FrameSourceTypes.Color | FrameSourceTypes.Depth);
    //En esta sección se deben seleccionar los canales a los que queremos tener acceso.

    // Vinculamos el método de la derecha al conjunto de métodos que se llamarán en caso de produc:
    _reader.MultiSourceFrameArrived += Reader_MultiSourceFrameArrived;
}
```

Cada vez que llegue un nuevo evento de la Kinect v2, extraeremos el *frame* a color y el *frame* de profundidad para procesarlos. En el caso del *frame* a color simplemente lo mandamos como argumento a la función que nos devolverá la matriz de píxeles de dicho *frame* mientras que en el caso del *frame* de profundidad, antes de poder devolver la matriz de píxeles, hemos de transformar la información de distancia que tiene cada pixel a su correspondiente intensidad de color en escala de grises.

Función que se ejecuta al detectar evento:

```
private void Reader_MultiSourceFrameArrived(object sender, MultiSourceFrameArrivedEventArgs e)
{
```

Detección del tipo de *frame* y llamada a *toBitmap* que nos devolverá la matriz de píxeles.

```
    // Color
    // Sentencia using, realiza la asignacion de frme y posteriormente
    using (var frame = reference.ColorFrameReference.AcquireFrame())
    {
        if (frame != null)
        {
            frame_array = ToBitmapAux(frame);
        }
    }

    // Profundidad
    using (var frameDepth = reference.DepthFrameReference.AcquireFrame())
    {
        if (frameDepth != null)
        {
            frame_arrayDepth = ToBitmapAuxDepth(frameDepth);
        }
    }
}
```


Por último, hemos de encolar el *frame* y crear el hilo del procesador que se encargará de ir desencolando y guardando en disco duro estos *frames* mientras que el programa principal seguirá recogiendo nuevos *frames* de la Kinect v2.

```
if (i == 0)
{
    Thread oThread = new Thread(new ThreadStart(Start));

    oThread.Start();
    while (!oThread.IsAlive) ;
}
if (i <= nSeconds * 30)
{
    Update(frame_array, true);
    i = i + 1;
}
```

Ha sido necesaria la creación del hilo pues el tiempo que tarda el procesador en transformar el *frame* a bmp y guardarlo en disco duro era mucho mayor que el tiempo en el que la Kinect v2 manda el siguiente *frame*, sin la implementación del hilo para paralelizar esta operación perdíamos muchos *frames*.

Mientras el programa principal sigue recogiendo y encolando *frames*, nuestro hilo se encarga de ir desencolando dichos *frames*.

```
public void Update(byte[] pixels, bool color)
{
    if (color == true)
        _queue.Enqueue(pixels);
    else
        _queueDepth.Enqueue(pixels);
}

public void Start()
{
    if (!IsRecording)
    {
        IsRecording = true;
    }

    while (IsRecording || _queue.Count > 0 || _queueDepth.Count > 0)
    {
        byte[] pixels;

        if (_queueDepth.TryDequeue(out pixels))
        {
            RecordBitmapDepth(pixels);
        }

        if (_queue.TryDequeue(out pixels))
        {
            RecordBitmap(pixels);
        }
    }
}
```


Una vez desencolado el *frame* se llama a la función encargada de transformarlo a bmp y guardarlo en el disco duro:

```
public void RecordBitmap(byte[] pixels)
{
    //widthG = 1920;
    //heightG = 1080;
    int stride = 1920 * format.BitsPerPixel / 8;

    BitmapSource frame3 = BitmapSource.Create(1920, 1080, 96, 96, format, null, pixels, stride);

    Bitmap bmp = new Bitmap(1920, 1080, System.Drawing.Imaging.PixelFormat.Format32bppPArgb);

    BitmapData data = bmp.LockBits(new System.Drawing.Rectangle(System.Drawing.Point.Empty, bmp.Size)
    frame3.CopyPixels(Int32Rect.Empty, data.Scan0, data.Height * data.Stride, data.Stride);
    bmp.UnlockBits(data);

    bmp.Save("C:\\Users\\sgr2\\Desktop\\frames\\frame" + f + ".bmp");

    f = f + 1;
    if (f == nSeconds * 30)
    {
        IsWriting = false;
    }

    bmp.Dispose();
}
```

El código completo del programa y de todas sus funciones está disponible para su consulta en el anexo.

Una vez grabados en el disco duro todos los *frames*, hemos de utilizar la secuencia de imágenes para crear el vídeo que formará parte del *dataset*. Se ha utilizado el software Adobe Premier Pro para realizar esta función.

Con la Kinect v2, se han grabado todos los *frames* obtenidos durante 20 segundos, como la cámara proporciona secuencias de imágenes a un ratio de 30 *frames* por segundo, cada vídeo se formará a partir de 600 imágenes.

Debido a que el algoritmo utilizado para obtener el ritmo cardíaco analiza *frame* a *frame* el vídeo, es necesario utilizar el modo progresivo a la hora de crearlo.

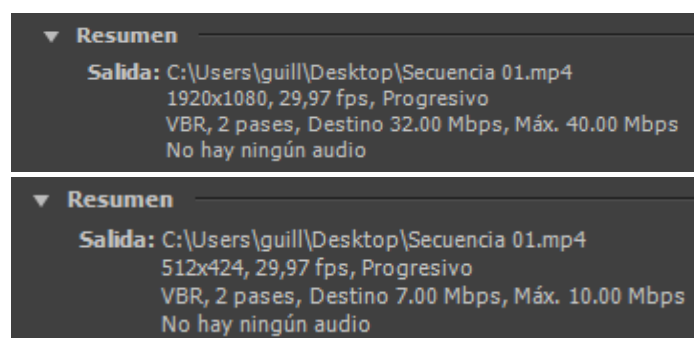


Figura 3-4: Resumen de las propiedades de los vídeos (color y profundidad) creados a partir de las imágenes obtenidas.

4 Evaluación

4.1 Introducción

En este apartado vamos a analizar los resultados obtenidos por nuestro algoritmo, para ello realizaremos una serie de pruebas con distintos vídeos. En nuestro caso, se realizarán las pruebas con el *dataset* generado (ver capítulo 3). Destacar que el *dataset* está formado por un conjunto de veinte vídeos con una separación de medio metro entre cada uno

Se proporcionarán los resultados obtenidos tras las pruebas, buscando conseguir un sistema de detección de ritmo cardíaco con la mayor precisión y fiabilidad posibles.

4.2 Marco de evaluación

4.2.1 Ground truth

El término *ground truth* es un término utilizado para hacer referencia al conjunto de datos proporcionados por la observación directa, datos objetivos, necesarios para verificar la precisión del conjunto de entrenamiento disponible.

Se ha grabado el ritmo cardíaco, en el momento de la grabación con la Kinect v2, con un pulsómetro (ver Figura 4-1) para poder comparar los resultados obtenidos del algoritmo con una buena referencia para verificar su precisión.



Figura 4-1: Pulsómetro utilizado para obtener el Ground truth.

4.2.2 Dataset

El *dataset*, MMMDPRvds (Multimodal Multidistance Pulse Rate video dataset) que se ha generado está compuesto por secuencias de vídeo de 20 segundos de duración cada una de ellas. Cada secuencia se ha grabado en 3 modalidades distintas: color procedente de la Kinect v2, profundidad procedente de la Kinect v2 y una secuencia de *ground truth* grabando el resultado obtenido del pulsómetro que llevaba puesto el sujeto grabado.

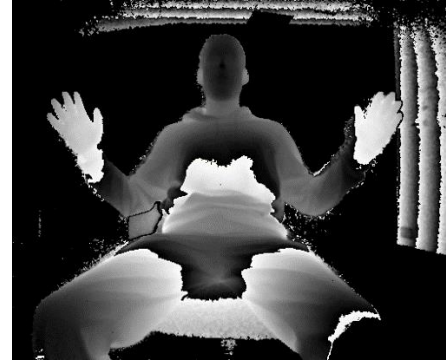
Estas grabaciones se han realizado con 2 voluntarios distintos y todas las grabaciones se han grabado en una sola sesión para cada voluntario.

Los recursos materiales utilizados para estas grabaciones son:

- Cámara Kinect v2 con tasa de 30 *frames* por segundo y resolución de 1920 x 1080 para las secuencias de color y 512x424 para las secuencias de profundidad.
- Pulsómetro CW Kalenji 100.



(a) RGB procedente Kinect v2



(b) Profundidad procedente Kinect v2

Figura 4-2: Secuencias de la Kinect v2 que componen el dataset

- PC con procesador Intel core i-5 con 8GB de memoria RAM y sistema operativo de 64 bits.

Se ha observado que, debido a la calidad de los frames proporcionados por la Kinect v2, los vídeos que componen el *dataset* ocupan mucho espacio en disco duro y resulta tedioso a la hora de compartirlos o desplazarlos de un lugar a otro, por ello surgió la idea de comprimirlos.

Para poder comprimir los vídeos hemos de asegurarnos que la compresión realizada no afecta en el rendimiento del algoritmo, para ello se ha realizado una comparación de los resultados arrojados por el algoritmo para secuencias de color (ver Tabla 4-1) tanto comprimidas como sin comprimir.

Se ha utilizado el códec H.264/MPEG-4 AVC para la compresión de las secuencias conservando siempre el *frame rate* y la resolución originales.

Distancia (m)	Comprimido (ppm)		Sin comprimir (ppm)		Error (%)	
	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2
0.5	97.5586	61.7432	98.2178	61.7432	0.67	0.00
1	61.3037	58.0078	61.3037	58.0078	0.00	0.00
1.5	44.6045	58.0078	44.6045	58.2275	0.00	0.37
2	72.2900	59.1064	72.2900	59.1064	0.00	0.00
2.5	56.2500	47.2412	56.0303	47.2412	0.39	0.00
3	82.1777	59.9854	82.1777	59.9854	0.00	0.00
3.5	69.4336	57.1289	69.6533	57.1289	0.31	0.00
4	64.1602	58.0078	64.1602	58.0078	0.00	0.00
4.5	63.5010	50.4297	63.7207	50.4297	0.34	0.00
5	55.8105	51.6357	55.8105	51.6357	0.00	0.00

Tabla 4-1: Error entre los resultados de secuencias comprimidas y sin comprimir

Como se puede comprobar en la tabla mostrada, la diferencia entre los resultados para las secuencias comprimidas respecto a las secuencias sin comprimir son muy bajas (0.00%-0.39%).

La diferencia en cuanto a espacio utilizado en disco duro entre ambas secuencias es muy considerable (2.61 GB para las secuencias sin comprimir y 114 MB para las comprimidas).

Basándonos en estos resultados se ha decidido utilizar las secuencias de vídeo comprimidas tanto para las secuencias de color como las secuencias de profundidad.

Para que se puedan apreciar mejor los resultados se ha realizado una representación visual de los resultados con Matlab (ver figura 4-3).

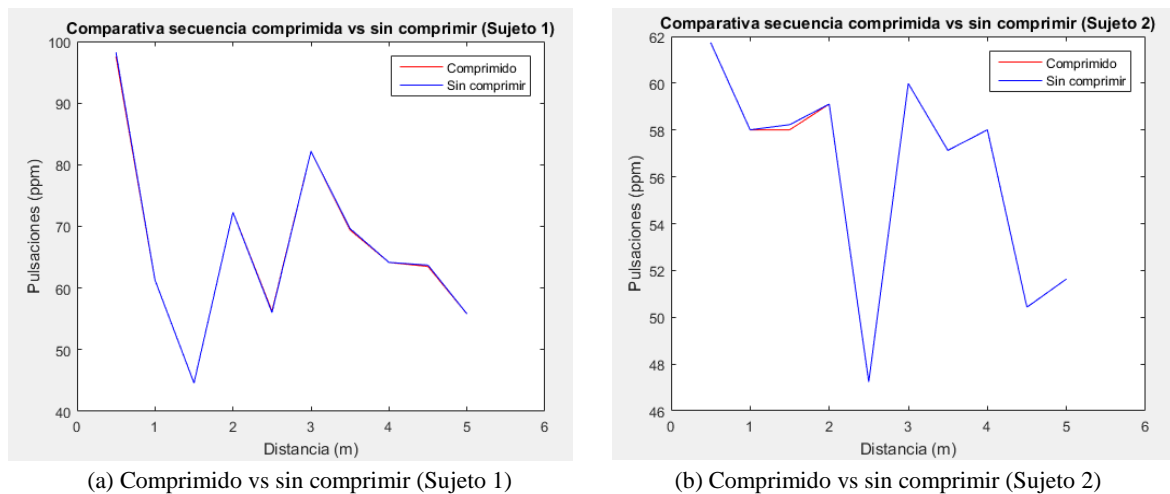


Figura 4-3: Visualización de los resultados para secuencias comprimidas respecto a las secuencias sin comprimir

4.2.3 Métrica

Para la evaluación de los algoritmos se han seleccionado las siguientes métricas usadas en [15].

- Media del pulso medido: Media del pulso medido en la duración de la secuencia en pulsaciones por minuto.
- Error: Error entre la media del pulso medido con el pulsómetro en la duración de la secuencia y la media del pulso resultante de aplicar el algoritmo a la secuencia.

4.3 Pruebas y Resultados

4.3.1 Análisis del algoritmo sobre secuencias en color del *dataset*

4.3.1.1 Análisis del algoritmo utilizando la transformada discreta de Fourier (DFT)

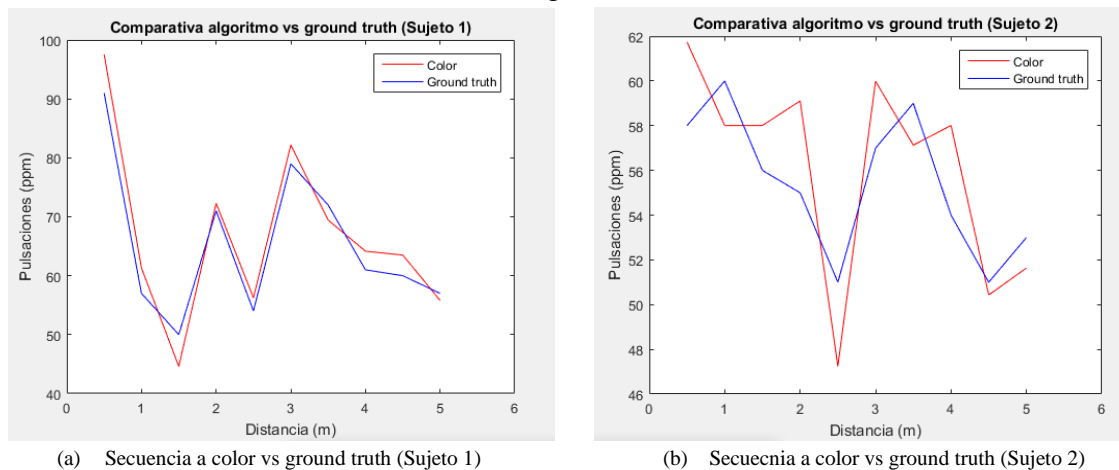
El objetivo de la siguiente prueba es comprobar el rendimiento del algoritmo para las secuencias a color proporcionadas por la Kinect v2 respecto a los resultados de *ground truth* obtenidos con el pulsímetro en función de la distancia del sujeto a la cámara. Para esta prueba se han utilizado todos los vídeos del *dataset* obtenido una vez comprimidos con el códec H.264.

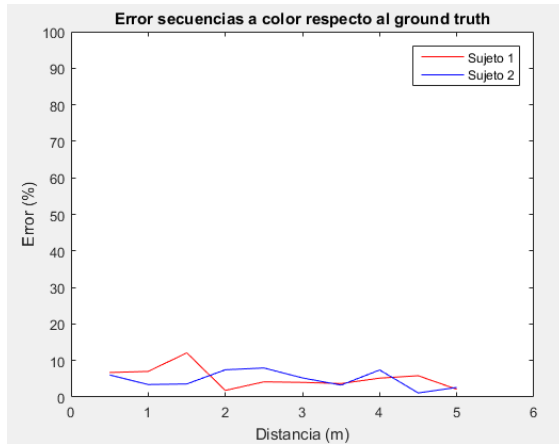
Distancia (m)	Color (ppm)		Ground truth (ppm)		Error (%)	
	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2
0.5	97.5586	61.7432	91	58	6.72	6.06
1	61.3037	58.0078	57	60	7.02	3.43
1.5	44.6045	58.0078	50	56	12.09	3.59
2	72.2900	59.1064	71	55	1.82	7.47
2.5	56.2500	47.2412	54	51	4.17	7.96
3	82.1777	59.9854	79	57	4.02	5.24
3.5	69.4336	57.1289	72	59	3.70	3.28
4	64.1602	58.0078	61	54	5.18	7.42
4.5	63.5010	50.4297	60	51	5.84	1.13
5	55.8105	51.6357	57	53	2.13	2.64

Tabla 4-2: Error entre los resultados de secuencias de color respecto al ground truth

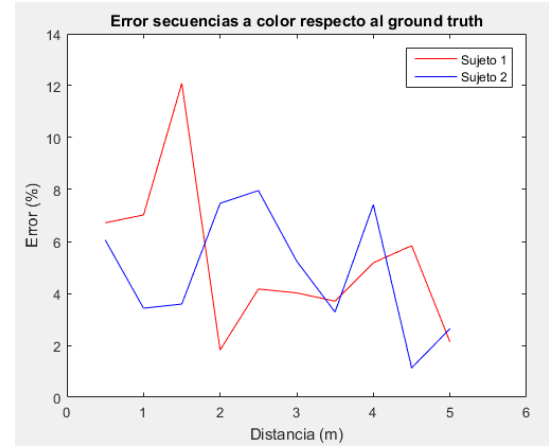
Analizando los resultados, se puede afirmar que, el algoritmo aplicado a las secuencias de color obtenidas por la Kinect v2, es muy fiable y arroja resultados muy parecidos a los obtenidos mediante el pulsímetro para todas las distancias medidas.

Para que se puedan apreciar mejor los resultados se ha realizado una representación visual de los resultados con Matlab (ver figura 4-4).





(c) Error respecto a ground truth de ambos sujetos sobre 100%



(d) Imagen ampliada sobre el eje de ordenadas

Figura 4-4: Visualización de los resultados para secuencias a color (DFT)

4.3.1.2 Análisis del algoritmo utilizando la densidad espectral de potencia (PSD)

A continuación, se han vuelto a analizar las secuencias a color aplicando el algoritmo a las mismas pero con la modificación de utilizar la densidad espectral de potencia (PSD) en lugar de la Transformada Discreta de Fourier (DFT).

Distancia (m)	Color (ppm)		Ground truth (ppm)		Error (%)	
	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2
0.5	115.1367	70.3125	91	58	26.52	21.23
1	76.2451	81.3984	57	60	33.76	35.66
1.5	52.2949	52.9541	50	56	4.59	5.75
2	52.9541	59.9854	71	55	34.08	9.06
2.5	59.9854	50.3174	54	51	11.08	1.36
3	54.0527	48.5596	79	57	46.15	17.38
3.5	53.8330	54.2725	72	59	33.75	8.71
4	51.6357	51.6357	61	54	18.14	4.58
4.5	51.4160	55.8105	60	51	16.70	9.43
5	51.6357	51.6357	57	53	10.39	2.64

Tabla 4-3: Error de secuencias de color respecto al ground truth utilizando la PSD.

A la vista de los resultados se puede apreciar que no arroja resultados tan precisos respecto al *ground truth* como la prueba realizada con la DFT.

Para que se puedan apreciar mejor los resultados se ha realizado una representación visual de los resultados con Matlab (ver figura 4-5).

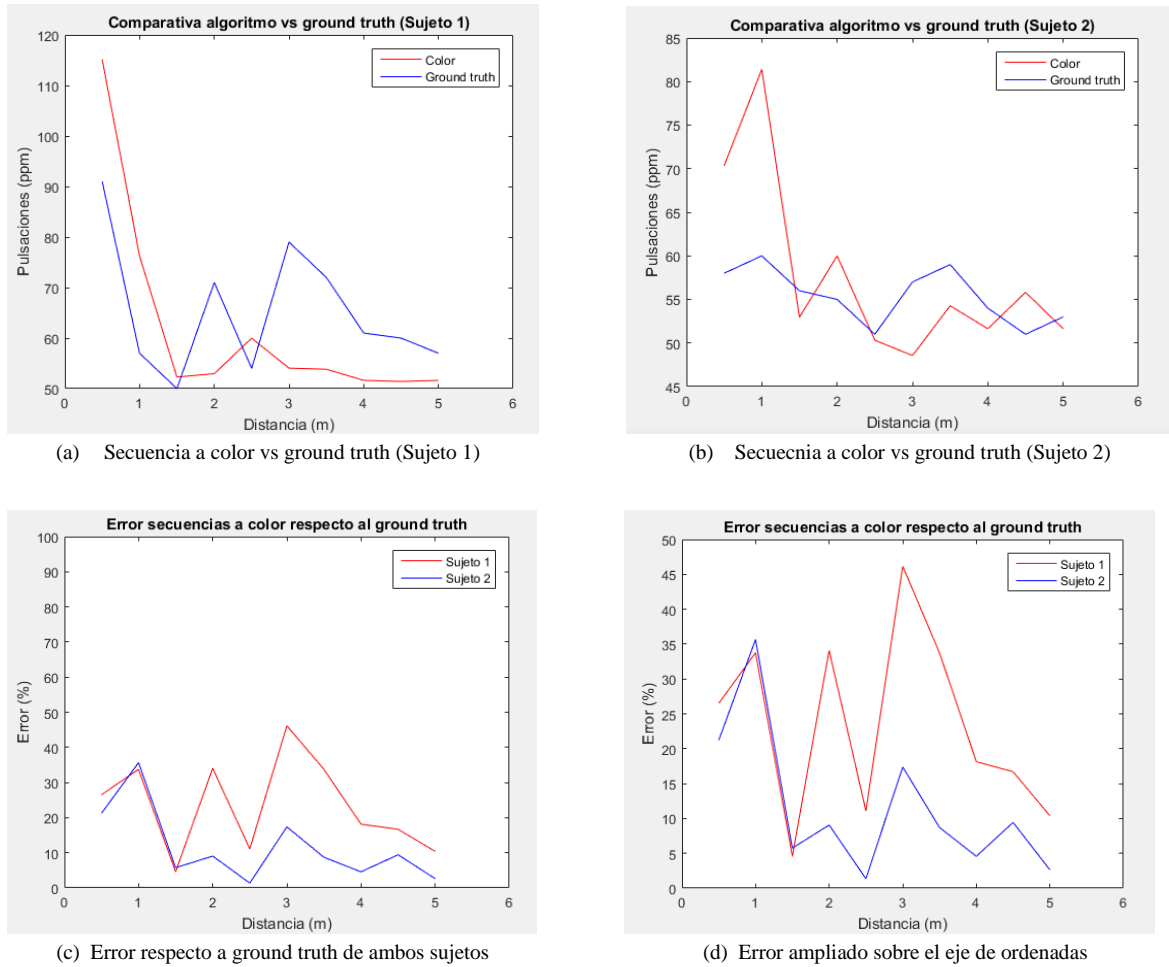


Figura 4-5: Visualización de los resultados para secuencias a color (PSD)

Se han representado también los resultados obtenidos utilizando la densidad espectral de potencia (PSD) junto con los resultados obtenidos utilizando la transformada discreta de Fourier (DFT) con respecto al *ground truth* para poder visualizar la diferencia en los resultados obtenidos, ver Figura 4-6.

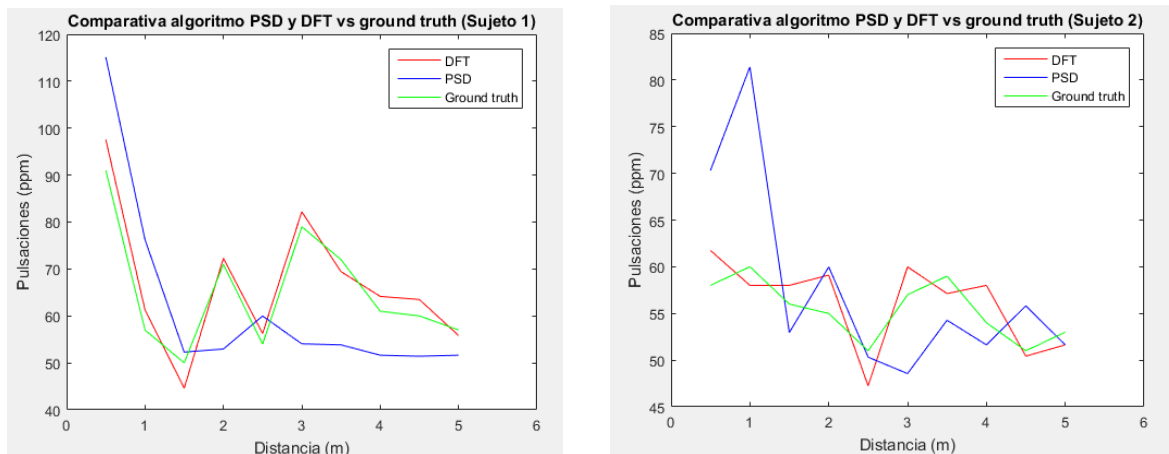


Figura 4-6: Resultados para secuencias a color (PSD vs DFT)

4.3.1.3 Análisis del algoritmo utilizando la frecuencia media (f_{media})

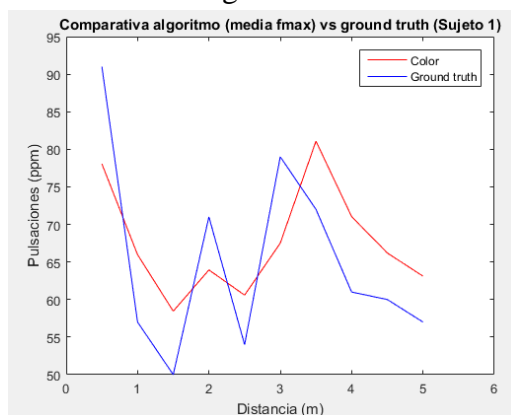
Utilizando las mismas secuencias de color, se han vuelto a obtener resultados utilizando la mejora propuesta para la función $find_max_energy$, como se comentó en el apartado anterior, ahora en lugar de utilizar la frecuencia a la que la energía es máxima con respecto a las cuatro señales transformadas obtenidas con las trayectorias de los puntos, se utilizará una media entre las frecuencias a cuya energía es máxima para cada señal.

Distancia (m)	Color (ppm)		Ground truth (ppm)		Error (%)	
	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2
0.5	78.0579	62.4573	91	58	16.58	7.69
1	65.9729	56.6895	57	60	15.74	5.84
1.5	58.4473	56.9641	50	56	16.89	1.72
2	63.9404	61.0291	71	55	11.04	10.96
2.5	60.5835	47.3511	54	51	12.15	7.71
3	67.5110	65.0940	79	57	17.02	14.2
3.5	81.0791	70.8120	72	59	12.61	20.02
4	71.0393	67.4561	61	54	16.46	24.92
4.5	66.1926	67.7856	60	51	10.32	32.91
5	63.1091	54.2725	57	53	10.72	2.41

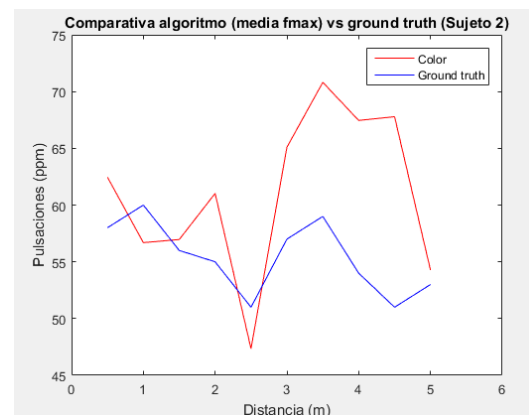
Tabla 4-4: Error entre los resultados de secuencias de color respecto al ground truth utilizando la media entre las cuatro frecuencias.

Analizando los resultados podemos afirmar que al utilizar la media entre las frecuencias obtenemos unos resultados con menos diferencia de error respecto al ground truth que si utilizamos únicamente la frecuencia a la que la energía se hace máxima no solo para cada componente sino comparando entre todas las componentes.

Se ha vuelto a realizar una representación gráfica para visualizar mejor los resultados ver Figura 4-7.



(a) Secuencia a color vs ground truth (Sujeto 1)



(b) Secuencia a color vs ground truth (Sujeto 2)

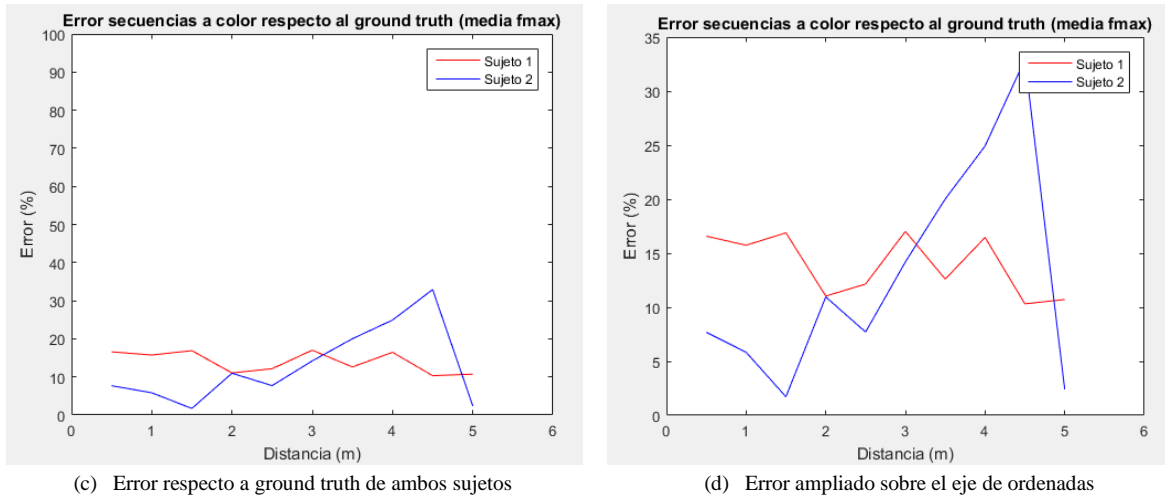


Figura 4-7: Visualización de los resultados para secuencias a color (media fmax)

Se han representado estos resultados obtenidos utilizando la media de las frecuencias respecto a los resultados arrojados por el algoritmo utilizando la frecuencia máxima respecto al ground truth para poder visualizar la comparación de los resultados para ambos métodos, ver Figura 4-8.

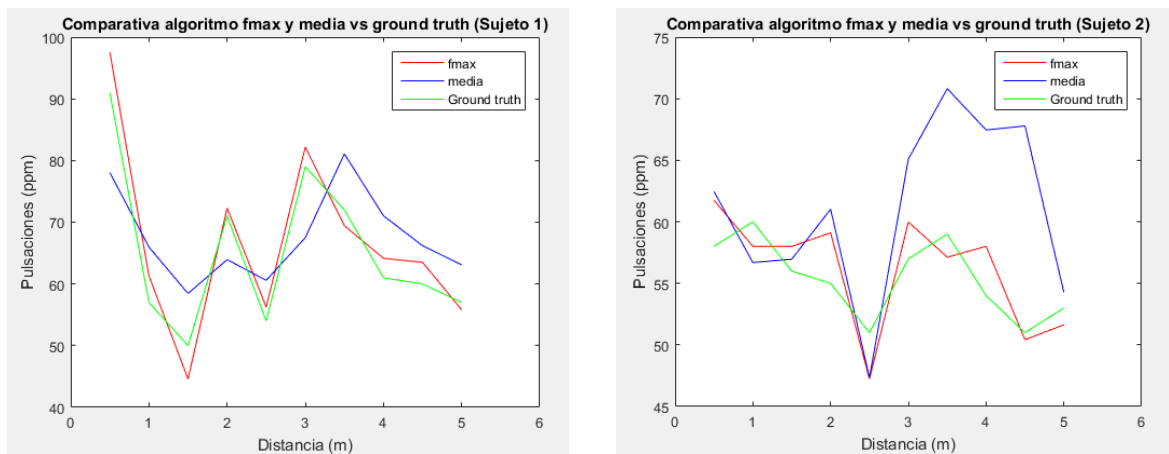


Figura 4-8: Resultados para secuencias a color (fmax vs media)

4.3.2 Análisis del algoritmo sobre secuencias de profundidad del dataset

4.3.2.1 Análisis del algoritmo utilizando la transformada discreta de Fourier (DFT)

Utilizando el mismo *dataset* se han realizado las mismas pruebas expuestas anteriormente, pero aplicando el algoritmo ahora a las secuencias de profundidad. Al haber grabado simultáneamente las secuencias de color y las de profundidad los datos de *ground truth* expuestos anteriormente son los mismos para estas secuencias.

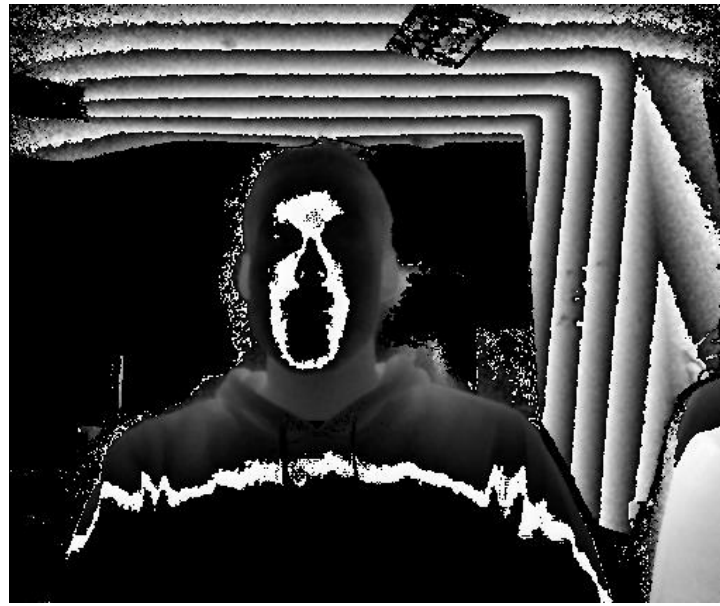
Distancia (m)	Profundidad (ppm)		Ground truth (ppm)		Error (%)	
	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2	Sujeto 1	Sujeto 2
0.5	127.2217	--	91	58	39.80	--
1	--	48.3398	57	60	--	24.12
1.5	--	51.8555	50	56	--	7.99
2	51.8555	51.8555	71	55	36.92	6.06
2.5	--	--	54	51	--	--
3	--	--	79	57	--	--
3.5	--	51.8555	72	59	--	13.78
4	51.8555	--	61	54	17.63	--
4.5	68.3350	60.8643	60	51	13.89	19.34
5	--	--	57	53	--	--

Tabla 4-5: Error entre los resultados de secuencias de profundidad respecto al ground truth

Como se puede observar, el algoritmo no arroja resultados para muchos de los casos y de los pocos en los que el algoritmo funciona, arroja el mismo resultado exacto para distintas distancias y sujetos distintos.

La razón por la que el algoritmo no funciona se debe a que no es capaz de detectar la región facial por lo que no selecciona las regiones de interés ni los puntos de interés. Analizando las imágenes en profundidad se ha descubierto que a ciertas distancias la Kinect v2 no proporciona *frames* fiables para el algoritmo.

Después de un análisis exhaustivo de las secuencias en profundidad que proporciona la Kinect v2 se ha descubierto que arroja *frames* fiables a unas distancias determinadas (ver Figura 4-5) mientras que en las secuencias comprendidas entre estas distancias sufren cambios de contraste en la región facial (ver Figura 4-7), a este rango de distancias en los que no funciona bien la cámara se le ha denominado “zonas muertas”. Al grabar alejándonos de la cámara a baja velocidad se aprecia como la imagen se va “adaptando” durante las zonas muertas hasta que muestra una imagen fiable y vuelve a empezar a adaptarse hasta que se alcanza la siguiente distancia en la que arroja una imagen fiable.



(a) Frame de profundidad en "zona muerta"



(b) Frame de profundidad en zona fiable

Figura 4-9: Comparación entre los frames de profundidad a distintas distancias

Una vez detectado y replicado el error y habiendo analizado las zonas muertas se han seleccionado las distancias a las que la Kinect v2 proporciona *frames* fiables (0.6m, 0.9m, 1.2m, 1.4m, 1.7m, 2.2m, 2.4m, 2.7m, 3.2m, 3.7m, 4m).

A diferencia de las secuencias a color, a partir de los 4 metros la Kinect v2 no proporciona ya una imagen de profundidad siendo éste el límite de rango del sensor.

Con estas distancias se ha generado un nuevo *dataset* volviendo a grabar a un solo sujeto a estas distancias y únicamente las secuencias en profundidad junto con los resultados del pulsímetro para poder comparar con el *ground truth*.

Distancia (m)	Profundidad (ppm)	Ground truth (ppm)	Error (%)
0.6	50.7568	53	4.42
0.9	51.1963	72	40.63
1.2	51.8555	70	34.99
1.4	50.0977	64	27.75
1.7	46.1426	65	40.87
2.2	60.3428	54	11.75
2.7	51.1963	55	7.43
3.2	52.5244	56	6.62
3.7	56.3936	61	8.17
4	65.6221	67	2.06

Tabla 4-6: Error entre los resultados de secuencias de profundidad respecto al ground truth (DFT)

Observando estos resultados se puede apreciar que el algoritmo aplicado a secuencias de profundidad no tiene la misma fiabilidad que cuando se aplica a secuencias de color. Se obtienen errores bastante considerables en el rango de distancias de 0.9m a 2.2m mientras que cuando estamos muy cerca de la cámara (0.6m) y a partir de los 2.2m hasta su límite de funcionamiento (4m), el algoritmo tiene un buen rendimiento, muy parecido al rendimiento que tiene con las secuencias de color.

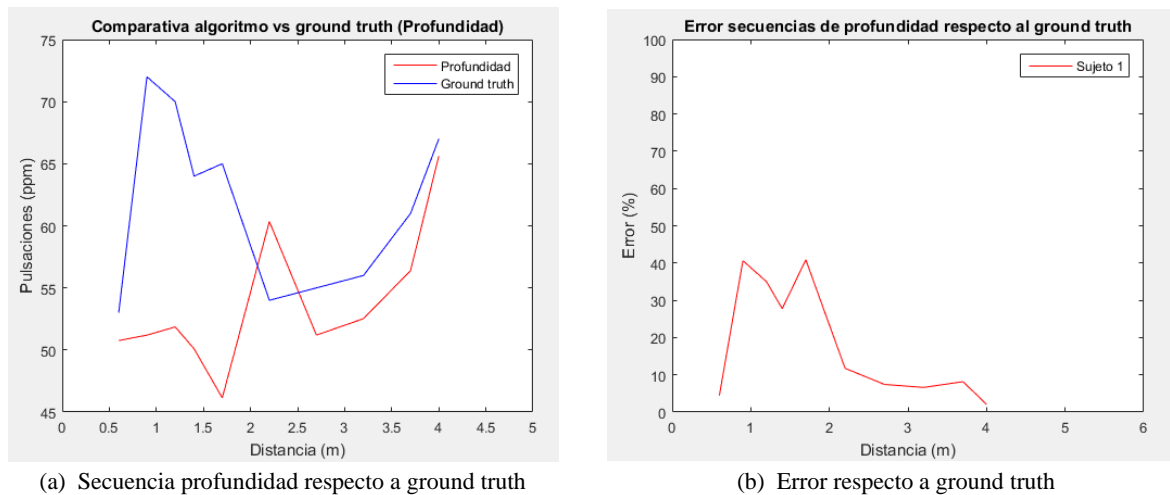


Figura 4-10: Visualización de los resultados para secuencias de profundidad (DFT)

4.3.2.2 Análisis del algoritmo utilizando la densidad espectral de potencia (PSD)

Con el nuevo *dataset* generado con las distancias óptimas para las secuencias de profundidad, se han vuelto a realizar las pruebas con las modificaciones realizadas al algoritmo para utilizar la PSD en lugar de la DFT y así comparar los resultados de nuevo y ver la diferencia en los resultados con las secuencias de profundidad.

Distancia (m)	Profundidad (ppm)	Ground truth (ppm)	Error (%)
0.6	55.3711	53	4.47
0.9	54.7119	72	31.60
1.2	49.6582	70	40.96
1.4	51.2012	64	24.99
1.7	54.0527	65	20.25
2.2	52.0752	54	3.70
2.7	53.4609	55	2.88
3.2	56.9092	56	1.62
3.7	63.3438	61	3.84
4	68.6670	67	2.49

Tabla 4-7: Error entre los resultados de secuencias de profundidad respecto al ground truth (PSD)

Analizando los resultados obtenidos se aprecia un comportamiento muy parecido al observado utilizando la DFT teniendo un comportamiento muy bueno a una distancia muy corta respecto a la cámara y dentro del rango de distancias de 2.2m a 4m.

A diferencia de los resultados obtenidos con las secuencias a color, cuando aplicamos la PSD en lugar de la DFT al algoritmo para las secuencias de profundidad, se obtiene un mejor resultado con una reducción considerable en el error para las zonas óptimas de operación del algoritmo.

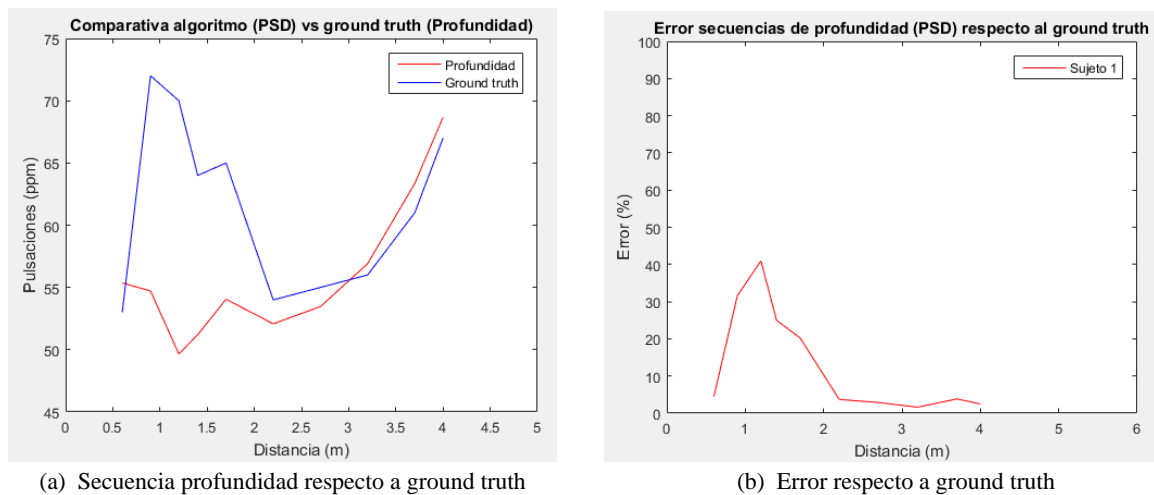


Figura 4-11: Visualización de los resultados para secuencias de profundidad (PSD)

A continuación, se representarán los resultados obtenidos utilizando la PSD junto con los obtenidos con la DFT para una mejor visualización de la comparación entre ambos resultados y la pequeña mejora obtenida al utilizar la densidad espectral de potencia.

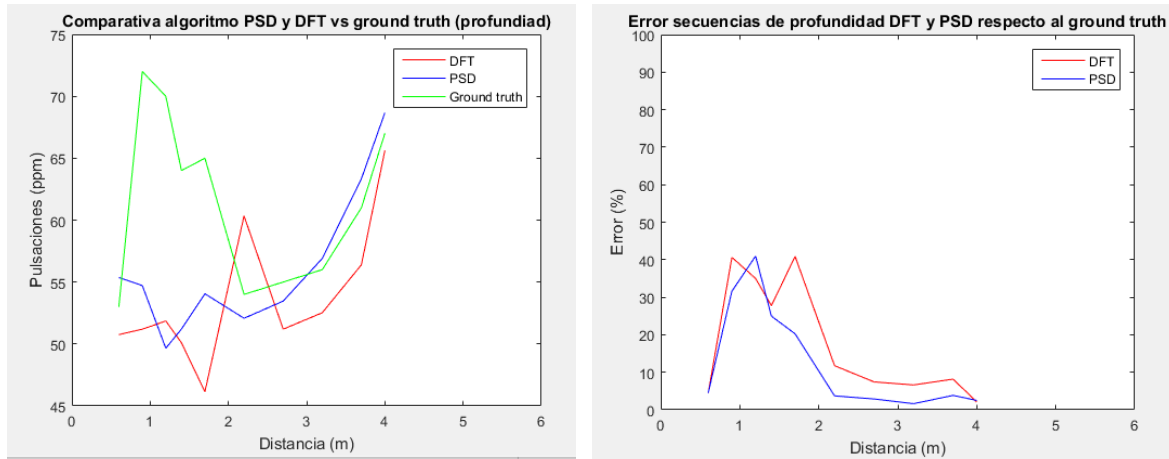


Figura 4-12: Resultados para secuencias de profundidad (PSD vs DFT)

4.3.2.3 Análisis del algoritmo utilizando la frecuencia media (f_{media})

Igual que se hizo con las secuencias de color, se va a volver a aplicar el algoritmo sobre este nuevo *dataset* pero aplicada la modificación del uso de la frecuencia media entre las cuatro componentes en lugar de utilizar sólo la frecuencia a la que la energía de la señal se hace máxima respecto a las cuatro señales.

Distancia (m)	Profundidad (ppm)	Ground truth (ppm)	Error (%)
0.6	52.0752	53	1.78
0.9	53.3171	72	35.04
1.2	51.8555	70	34.99
1.4	57.0740	64	12.14
1.7	54.3274	65	19.64
2.2	53.7659	54	0.44
2.7	54.5198	55	0.88
3.2	55.9482	56	0.09
3.7	62.4579	61	2.39
4	65.7143	67	1.96

Tabla 4-8: Error entre los resultados de secuencias de profundidad respecto al ground truth (media f_{max})

A la vista de los resultados obtenidos podemos apreciar una mejora considerable en el rendimiento del algoritmo al utilizar la media entre las frecuencias obtenidas en lugar de utilizar la frecuencia a la que la energía se hace máxima respecto a las cuatro componentes, obtenemos un error mucho menor y unos resultados casi exactos a los medidos con el pulsómetro.

A continuación, se representan los resultados con respecto al ground truth para una mejor visualización de los mismos, ver Figura 4-13.

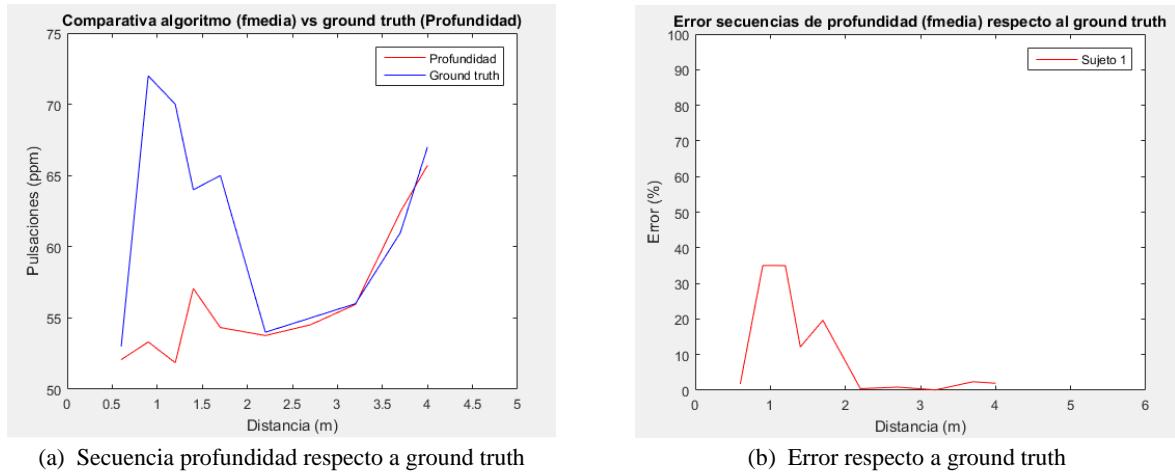


Figura 4-13: Visualización de los resultados para secuencias de profundidad (fmedia)

También se han representado los resultados respecto a los resultados iniciales del algoritmo sin modificar para visualizar al diferencia en los resultados con el cambio realizado, ver Figura 4-14.

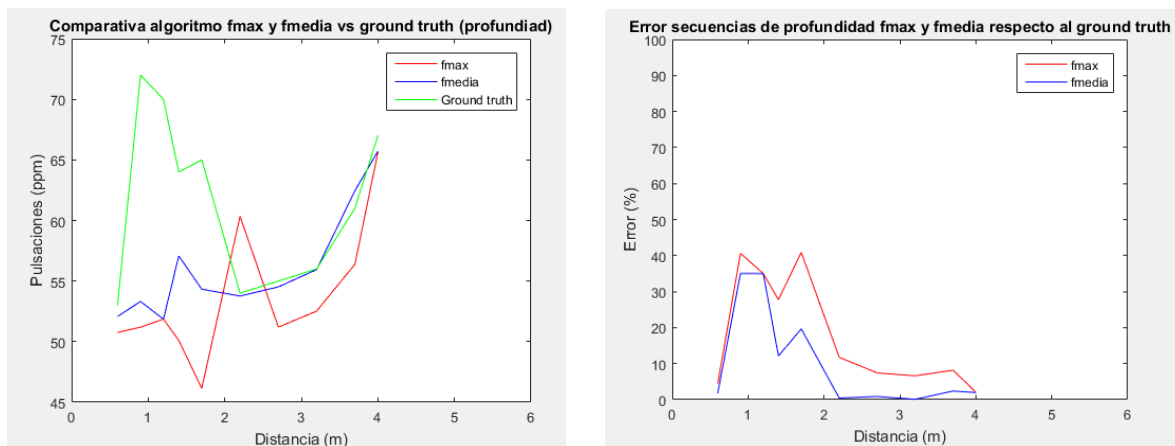


Figura 4-14: Resultados para secuencias de profundidad (fmax vs fmedia)

Como se puede observar en las gráficas, el uso de la media de las frecuencias de las cuatro componentes mejora los resultados obtenidos alcanzando unos valores casi exactos a los del ground truth para el rango de funcionamiento de 0.6m y 2.2m a 4m.

A partir de los 2.2m nos encontramos con un error máximo de 2.39% y con varios valores por debajo del 1% de error.

Cabe destacar que en las zonas de mal funcionamiento del algoritmo (0.7m – 2.2m) también se ha reducido el error de forma considerable.

4.4 Conclusiones

Después de realizar las pruebas pertinentes en las secuencias de vídeo disponibles y analizando los resultados obtenidos en la evaluación, podemos afirmar que el sistema ofrece un buen rendimiento en detección de ritmo cardíaco para secuencias de color.

Ofrece buen resultado cuando se utilizan secuencias de profundidad siempre y cuando las secuencias sean grabadas en las zonas fiables del sensor y estén comprendidas en el rango de 2.2m a 4m y para las secuencias grabadas muy cerca de la cámara (0.6m).

A la vista de los resultados obtenidos y comparando con el estudio realizado sobre este algoritmo con la Kinect v1 se puede determinar que existe una mejora considerable en el rendimiento del algoritmo debido a la calidad y aumento de resolución de las secuencias que proporciona la Kinect v2.

Con respecto a las mejoras implementadas, tanto el uso de la PSD en lugar de la transformada discreta de Fourier como el uso de la frecuencia media en lugar de la frecuencia a la que la energía es máxima no producen una mejora en las secuencias de color, pero mejoran considerablemente los resultados obtenidos en las secuencias de profundidad, funcionando mejor que el algoritmo original.

Cabe destacar que, a pesar de las mejoras, el algoritmo para secuencias de profundidad no funciona bien fuera del rango de 2.2m a 4m o muy cerca de la cámara (0.6m).

Para apreciar de una forma global las mejoras obtenidas en las secuencias de profundidad al utilizar la densidad espectral de potencia (PSD) y la frecuencia media, se ha realizado una tabla comparativa de los errores obtenidos junto con su respectiva representación en Matlab (ver Tabla 4-9 y Figura 4-15).

Distancia (m)	DFT, error (%)	PSD, error (%)	Fmedia, error (%)
0.6	4.42	4.47	1.78
0.9	40.63	31.60	35.04
1.2	34.99	40.96	34.99
1.4	27.75	24.99	12.14
1.7	40.87	20.25	19.64
2.2	11.75	3.70	0.44
2.7	7.43	2.88	0.88
3.2	6.62	1.62	0.09
3.7	8.17	3.84	2.39
4	2.06	2.49	1.96

Tabla 4-9: Comparativa errores obtenidos para los distintos métodos utilizados

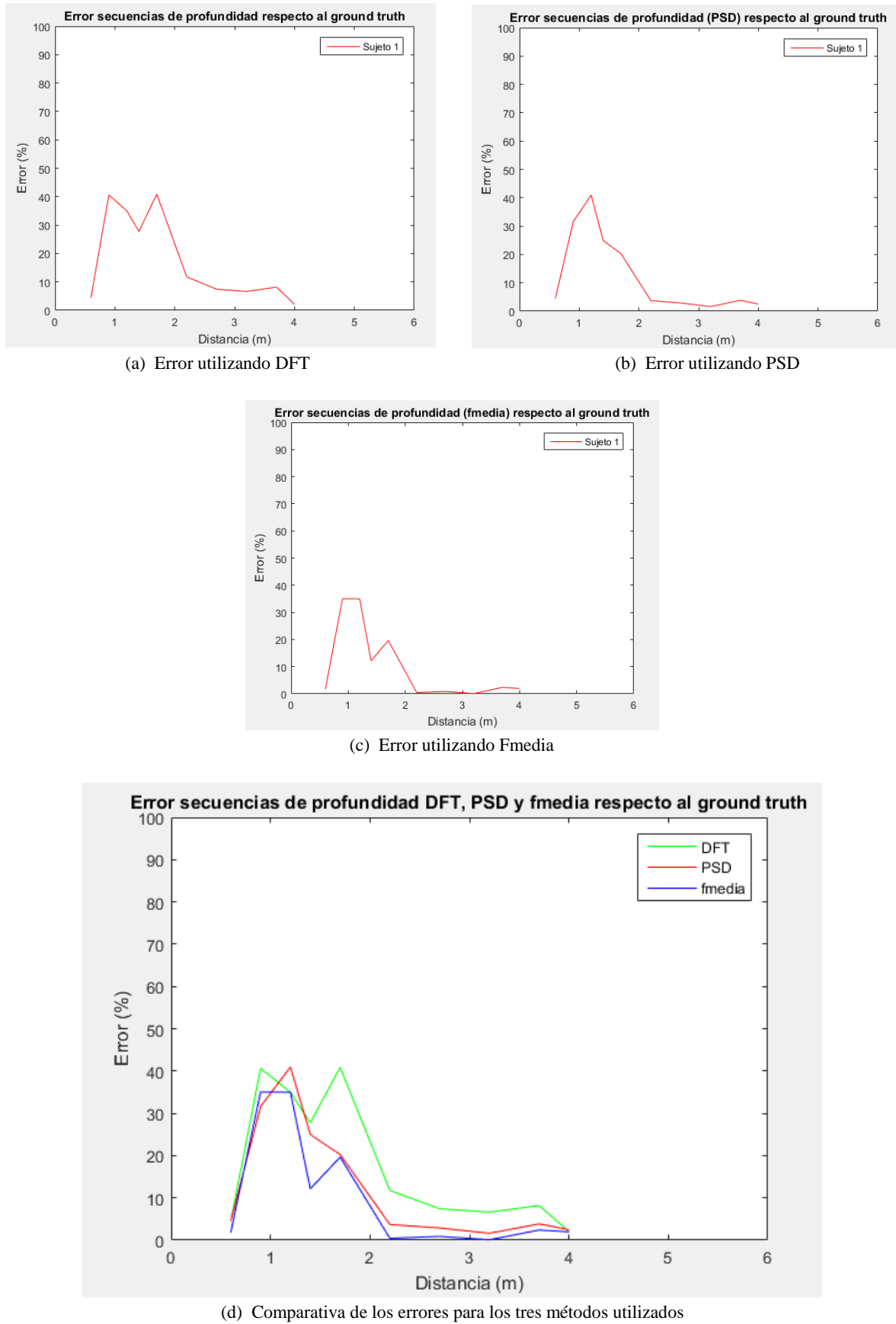


Figura 4-15: Visualización errores para los tres métodos utilizados

5 Conclusiones y trabajo futuro

5.1 Conclusiones

El objetivo principal de este proyecto era adaptar y mejorar el algoritmo, diseñado por el MIT e implementado para la Kinect v1 por Erik Velasco, a la Kinect v2 para obtener el ritmo cardíaco de una persona mediante los micromovimientos producidos en la región facial debidos al flujo sanguíneo.

Otro de los objetivos del proyecto era el diseño y desarrollo de un programa capaz de extraer las secuencias de color y profundidad proporcionadas por la Kinect v2 creando así un *dataset* que sirviese para la evaluación de los algoritmos y que pueda ser utilizado para posibles proyectos derivados de éste.

Después del análisis realizado en este documento de las distintas pruebas realizadas podemos afirmar que el objetivo se ha cumplido de forma razonable. Se han cumplido los objetivos de adaptación para las secuencias tanto de color como de profundidad teniendo éstas un rendimiento inferior a las secuencias de color.

También se ha analizado la viabilidad del algoritmo respecto a la distancia, una vez analizado el comportamiento del algoritmo se ha visto que para las secuencias de color el algoritmo funciona en la totalidad de las distancias utilizadas mientras que para las secuencias de profundidad el rendimiento del algoritmo baja considerablemente cuando trabajamos fuera de los rangos de distancia de 2.2m–4m y 0m-0.6m.

El uso de la densidad espectral de potencia en sustitución a la transformada discreta de Fourier proporciona una mejora al algoritmo para las secuencias de profundidad.

La modificación del algoritmo de utilizar la media de las cuatro frecuencias obtenidas debido al máximo de energía de las DFT's de cada componente en lugar de utilizar una única frecuencia correspondiente a la máxima energía entre las cuatro componentes ha supuesto una mejora muy considerable para las secuencias de profundidad funcionando casi a la perfección con errores de 0.09%-2%.

5.2 Trabajo futuro

A la vista de los resultados obtenidos en este proyecto se propone trabajar en:

- Profundizar en las secuencias de profundidad e intentar mejorar el rendimiento del algoritmo fuera de los rangos de 0m-0.6m y 2.2m-4m.
- Ampliar el *dataset* con secuencias grabadas con el mismo sujeto con diferencias notables en el ritmo cardíaco para ver el comportamiento del algoritmo con un rango de pulsaciones más extenso.

- Implementar un sistema de medición en tiempo real que no necesite grabar los *frames* en el disco duro sino que vaya analizando los *frames* según se van obteniendo, para ello se debería implementar una interfaz de usuario.

6 Referencias

- [1] Inglis B., “Primeras civilizaciones”, en Inglis B., ed., *Historia de la medicina*, Barcelona, Grijalbo, 1968, pp. 21-32.
- [2] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman, Eulerian video magnification for revealing subtle changes in the world, *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1-8, 2012.
- [3] J. L. I. Pavlidis, Thermal image analysis for polygraph testing, *IEEE Engineering in Medicine and Biology Magazine*, vol. 21, pp. 56-64, 2002.
- [4] C. Takano and Y. Ohta, Heart rate measurement based on a time-lapse image., *Medical engineering & physics*, vol. 29, pp. 853-857, Oct. 2007.
- [5] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 511-518, 2001.
- [6] A. Kleiner and D. Rabinowitz, *Video Heart Rate Detection*, 2014.
- [7] M. Z. Poh, D. J. McDu, and R. W. Picard, Advancements in noncontact, multiparameter physiological measurements using a webcam, *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 1, pp. 7-11, 2011.
- [8] G. R. Tsouri, S. Kyal, S. Dianat, and L. K. Mestha, Constrained independent component analysis approach to nonobtrusive pulse rate measurements, *Journal of biomedical optics*, vol. 17, no. 7, pp. 0770111-0770114, 2012.
- [9] T. Fitzpatrick, Sun and skin (soleil et peau), *J Med Esthétique*, vol. 2, pp. 33-34, 1975.
- [10] F. Bousefsaf, C. Maaoui, and A. Pruski, Continuous wavelet ltering on webcam photoplethysmographic signals to remotely assess the instantaneous heart rate, *Biomedical Signal Processing and Control*, vol. 8, pp. 568-574, Nov. 2013.
- [11] C. Takano and Y. Ohta, Heart rate measurement based on a time-lapse image., *Medical engineering & physics*, vol. 29, pp. 853-857, Oct. 2007.

- [12] L. Carvalho, M. Virani, and M. Kutty, Analysis of Heart Rate Monitoring Using a Webcam, *Analysis*, vol. 3, no. 5, pp. 6593-6595, 2014.
- [13] S. Kwon, H. Kim, and K. S. Park, Validation of heart rate extraction using video imaging on a built-in camera system of a smartphone, *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 2174-2177, 2012.
- [14] J. B. Bolkhovsky, C. G. Scully, and K. H. Chon, Statistical analysis of heart rate and heart rate variability monitoring through the use of smart phone cameras, in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pp. 1610-1613, IEEE, 2012.
- [15] G. Balakrishnan, F. Durand, and J. Guttag, Detecting pulse from head motions in video, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3430-3437, June 2013
- [16] J. Shi and C. Tomasi, Good features to track, in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pp. 593-600, IEEE, 1994.
- [17] B. D. Lucas, T. Kanade, et al., An iterative image registration technique with an application to stereo vision., in *IJCAI*, vol. 81, pp. 674-679, 1981.
- [18] R. Irani, K. Nasrollahi, and T. B. Moeslund, Improved Pulse Detection from Head Motions Using DCT, *International Conference on Computer Vision Theory and Applications*, p. 8, 2014.
- [19] Erik Velasco Salido, Detección de ritmo cardíaco mediante vídeo, Trabajo Fin de Grado, Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Junio 2015.

Anexo A: Presupuesto

- 1) Ejecución Material**
 - Compra de ordenador personal (Software incluido)..... 2.000 €
 - Material de oficina..... 200 €
 - Total de ejecución material..... 2.200 €
- 2) Gastos generales**
 - 16 % sobre Ejecución Material..... 352 €
- 3) Beneficio Industrial**
 - 6 % sobre Ejecución Material..... 132 €
- 4) Honorarios Proyecto**
 - 640 horas a 15 € / hora..... 9600 €
- 5) Material fungible**
 - Gastos de impresión..... 60 €
 - Encuadernación..... 200 €
- 6) Subtotal del presupuesto**
 - Subtotal Presupuesto..... 12060 €
- 7) I.V.A. aplicable**
 - 21% Subtotal Presupuesto 2532.6 €
- 8) Total presupuesto**
 - Total Presupuesto..... 14593,6 €

Madrid, Junio de 2016

El Ingeniero Jefe de Proyecto

Fdo.: Guillermo Guillén Mier

Ingeniero de Telecomunicación

Anexo B: Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de *un sistema de detección de ritmo cardíaco mediante vídeo-monitorización*. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado “Presupuesto de Ejecución Material” que hoy designa otro concepto.

Condiciones particulares

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.